อัลกอริทึมเลขคณิตพื้นฐานเชื่อมตรงสำหรับระบบเวกเตอร์สามมิติ

นายสราวุฒ รังสรรค์วิจิตร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2547

ISBN 974-17-5806-5

ON-LINE FUNDAMENTAL ARITHMETIC ALGORITHMS FOR THREE-DIMENSIONAL VECTOR SYSTEM

Mr. Saravut Rangsunvigit

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2004
ISBN 974-17-5806-5

Thesis Title	ON-LINE FUNDAMENTAL ARITHMETIC ALGORITHMS FOR THREE-DIMENSIONAL VECTOR SYSTEM.
By	SARAVUT RANGSUNVIGIT
Field of study	COMPUTER SCIENCE
Thesis Advisor	ATHASIT SURARERKS, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master 's Degree

………………………………………… Dean of Faculty of Engineering

(Professor Direk Lavansiri, Ph.D.)

THESIS COMMITTEE

…………………………………………… Chairman

(Associate Professor Prabhas Chongstitvatana, Ph.D.)

…………………………………………… Thesis Advisor

(Athasit Surarerks, Ph.D.)

…………………………………………….. Member

(Arthit Thongtak, Ph.D.)

……………………………………………… Member

(Assistant Professor Arnon Rungsawang, Ph.D.)

นายสราวุฒ รังสรรค์วิจิตร : อัลกอริทึมเลขคณิตพื้นฐานเชื่อมตรงสำหรับระบบเวกเตอร์สามมิติ. ((ON-LINE FUNDAMENTAL ARITHMETIC ALGORITHMS FOR THREE-DIMENSIONAL VECTOR SYSTEM) อ. ที่ปรึกษา : อาจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์ 49 หน้า. ISBN 974-17-5806-5

งานวิจัยนี้เป็นการเสนอระบบแทนเวกเตอร์สามมิติแบบใหม่ ระบบที่เสนอนี้ประกอบด้วย เลขฐานที่เป็นจำนวนเต็ม และ เซตจำกัดของดิจิตที่เป็นเวกเตอร์ที่มีเครื่องหมาย งานวิจัยนี้ยังได้เน้นไปที่การคำนวณเชื่อมตรงผสานเข้ากับแนวคิดของการทำงานท่อตรง คุณสมบัติที่สำคัญสำหรับระบบการทำงานเชื่อมตรงได้ถูกศึกษาในงานนี้เช่นเดียวกัน นั่นคือความหน่วงเชื่อมตรง ซึ่งเป็นจำนวนของดิจิตที่น้อยที่สุดของตัวถูกดำเนินการที่ต้องใช้ในการผลิตดิจิตแรกของผลลัพธ์ เพื่อให้การทำงานเชื่อมตรงทำได้ ระบบแทนเวกเตอร์จะต้องมีคุณสมบัติซ้ำซ้อน หมายความว่า เวกเตอร์ใดๆ สามารถหารูปแบบแทนได้มากกว่าหนึ่งรูปแบบ แนวคิดของระบบจำนวนมีเครื่องหมายจึงถูกนำมาประยุกต์ใช้ในงานนี้ด้วย ในระบบแทนเวกเตอร์แบบใหม่นี้ การคำนวณพื้นฐานที่ใช้สำหรับเวกเตอร์สามมิติ อันได้แก่ การบวก การลบ และการคูณ (ครอสโปรดัก) ได้ถูกแสดงให้เห็นว่าสามารถทำงานได้ด้วยการทำงานที่คล้ายกับระบบจำนวนตัวเลขแบบคลาสสิค.

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต...
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา.......................................
ปีการศึกษา 2547

#4671438621 : MAJOR COMPUTER SCIENCE

KEY WORD: THREE DIMENSIONAL VECTOR SYSTEM/ REDUNDANT NUMBER SYSTEM/ ON-LINE ARITHMETIC

SARAVUT RANGSUNVIGIT: ON-LINE FUNDAMENTAL ARITHMETIC ALGORITHMS FOR THREE-DIMENSIONAL VECTOR SYSTEM. THESIS ADVISOR: ATHASIT SURARERKS, Ph.D., 49p. ISBN 974-17-5806-5

This research is to introduce a novel three-dimensional vector representation system. The proposed system composes of an integer base and a finite set of signed-vector-digit. The research is also focused on an on-line computation mode combining with a pipelining concept. An important characteristic for an on-line system is also studied, that is the on-line delay, the smallest integer which is the number of digits of the inputs used for producing the first digit of the output. In order to do this, the vector representation system should be a redundant system which means any vector can have more than one finite representation. The concept of signed digit number system is applied to this work. Using the new representation, some fundamental arithmetic operations for three-dimensional vector such as addition, subtraction, and multiplication (cross product) are shown to be realized, similar to the classical number system.

Department Computer Engineering Student's signature..................................

Field of study Computer Science Advisor's signature..................................

Academic year 2004

ACKNOWLEDGEMENTS

This thesis cannot be completely finished without the great help from my thesis advisor, Dr. Athasit Surarerks who has chosen me to be one of his students in ELITE group. I would like to take this special opportunity to send my respect and thank you for the great support, suggestion, and guidance to him. He has always given advice and spent time with me days and nights since I have started my research through the completion. Even though I had not been around in Thailand during my thesis composing, because of the magnificent help and support from him, I can finish the thesis on time. I am very thankful to have him as my thesis advisor.

I also would like to thank you all of my thesis committees, Associate Professor Dr. Prabhas Chongsatitvatana , Dr. Arthit Thongtak, and Assistant Professor Dr. Arnon Rungsawang who have taken precious time to give useful comments, suggestions and guidances in order to get this thesis to be more completed.

Finally, I would like to thank you my parents, brothers, and sister who have given a support and encouragement to me for taking this master degree program in Chulalongkorn University until I could finish.

CONTENTS

page

ABSTRACT (THAI) iv
ABSTRACT (ENGLISH) v
ACKNOWLEDGEMENTS vi
CONTENTS vii
LIST OF TABLES viii
LIST OF FIGURES ix
CHAPTER

I INTRODUCTION 1
II PRELIMINARIES 7
2.1 Number Systems 7
2.1.1 Signed digit number system 8
2.1.2 Negative integer base number system 10
2.1.3 Complex number 10
2.1.4 Three-dimensional vector system 12
2.2 On-line Arithmetic 15
2.2.1 On-line addition 16
2.2.2 On-line subtraction 18
2.2.3 On-line multiplication 19
III THREE-DIMENSIONAL VECTOR REPRESENTATION 21
3.1 Introduction 21
3.2 The representation 22
3.3 Completeness 23
3.4 Redundancy 25
IV ON-LINE ADDITION/SUBTRACTION FOR THREE-DIMENSIONAL VECTOR REPRESENTATION 27
4.1 On-line addition 27
4.2 On-line subtraction 34
4.3 On-line multiplication 36
V CONCLUSION 49
REFERENCES 50
BIOGRAPHY 51

LIST OF TABLES

Table page

1.1 Binary number system table (n = 0-10) ……..………………………………. 2
2.1 On-line multiplication of A and B where A = B = $(.0011011111)_2$ …………. 20
4.1 On-line addition of 60i + 23j + 32k and 48i + 36j + 19k …………………. 33
4.2 On-line subtraction of 50i + 13j + 22k and 38i + 26j + 9k …………………. 36
4.3 On-line multiplication of 84i+27j+13k and 37i + 93j + 82k following the algorithm M_v ……………………………………..…………………. 46

LIST OF FIGURES

Figure page

2.1 The representation of 14 in the number system (2, {-1, 0, 1}) ………………. 10

2.2 Complex number representation on two-dimensional plane ………….…….10

2.3 The representation of three-dimensional vector system ……………….……. 13

2.4 Cross product calculation of three-dimensional vector system based on righthand finger Concept ……………………………….………….……. 14

2.5 The addition of 708 and 766 in the system (5, {-3, -2, -1, 0, 1, 2, 3 })………. 17

2.6 The subtraction of 766 from 708 in the system (5, {-3, -2, -1, 0, 1, 2, 3 }) …. 18

4.1 The relation of the inputs, $r_{j,t-\delta}$, $s_{j,t-\delta}$, and the output ($z_{j,t}$)……………………..…. 30

4.2 The process of an on-line multiplication algorithm M_v…………………………......... 43

4.3 On-line multiplication of 84i+27j+13k and 37i+93j+82k ……………………. 45

CHAPTER I

INTRODUCTION

The number has a long history for human being. It has been discovered for a long period of time. Number representation can be composed by either finite or infinite set of strings normally represented by alphabets or digits. In arithmetic computation, each representation can be used to do any source of computation by particular algorithms. Any real or complex number can be represented by a set of strings which are alphabets or digits as described earlier.

$$(X_a X_{a-1} X_{a-2} X_{a-3} X_{a-4} \ldots .)_n = X_a n^a + X_{a-1} n^{a-1} + X_{a-2} n^{a-2} + X_{a-3} n^{a-3} + \ldots,$$

where a is positional number of each digit and n is a base number which can be both a real and complex number. This kind of representation is called *ß-repesentation*.

The binary number system where its base number is 2 is widely used in the computer arithematic world. It generally contains a sequence of 0s and 1s.

$b_n\, b_{n-1} \ldots\, b_2 b_1 b_0$ represents the number

$$b_n 2^n\; b_{n-1} 2^{n-1} + \ldots + b_2 2^2\; b_1 2^1\; b_0 2^0$$

Basically, any base ten number can be represented as a summation of power of two as shown in Table 1.1

In term of binary computation, the addition of any two binary number system can be done following the least significant digit first (LSDF) mode which is the computation to flow from right to left as it normally does in the decimal base number system. The concept of propagation (carry over) is applied in order to make sure that each output result is still having the valid format of binary number system in which only 0 and 1 are allowed.

The multiplcation of any binary number system has the same calculation steps as the decimal number system. The concept of addition of binary number system is also applied for multiplication operation as well.

In addition to a positive base number system that is often used, a negative integer base number can be used to represent a number system without a sign as well.

Table 1.1 : *Binary number system table (n = 0-10)*

N	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

Example 1.1 *Find the addition of 81 and 25 in base number –2 with the digit set as {0,1},*

Solution

$81 = (1010001)_{-2}$

$25 = (1101001)_{-2}$

The addition of 81 and 25 in the base –2 number system can be done as follows:

$$\begin{array}{rcccccccc} 81 & = & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ & & & & & & & & + \\ 25 & = & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ \hline & & 2 & 1 & 1 & 1 & 0 & 0 & 2 \end{array}$$

Number 2 can be represented as 110 in the base –2 number system, the above example answer could be further calculated as below:

$$(110111110)_{-2} = -2 + 4 - 8 - 32 - 128 + 256 = 106$$

A complex number contains both real and imaginary part. It was firstly known in 1st century AD by Greek mathematician, Heron of Alexandria. Until during the mid of 19th century, Willian Rowan Hamiltion discovered the representation of a complex number as a+bi on the xy plane where both a and b are a real number. Some complex number representation have beed examined in [1,2].

Such a fascinated number can also be pointed in a two-dimensional plane with taking 'a' value on the x-axis and 'b' value of the y-axis. The complex number development is still going along in the 20th century. It has also been adapted to be used in a modern world such as in both electrical and mechanidal engineering field.

Two-dimensional vector can also be reprsented using a complex number, one dimension corresponds to a real part and the other dimension corresponds to an imaginary part. Fundamental arithmetic operations, such as addition, subtraction, dot product and cross product, can also be performed in the complex number system.

This concept of vector representation using complex number system is extented to a multi-dimensional vector system. A vector is usually described by a summation of products. A unit vector for each dimension is proposed. Any n-dimensional vector $X = (x_1, x_2, x_3, \ldots, x_n)$ can be expressed by

$$x_1 \times \mathrm{u}_1 + x_2 \times \mathrm{u}_2 + x_3 \times \mathrm{u}_3 + \ldots + x_n \times \mathrm{u}_n,$$

where u_i is a unit vector for the i^{th} dimension. The value of the expression is normally called a *vector value* of the representation. By the same way as two-dimensional vector system, fundamental arithmetic operations for vector system can also be performed in the multi-dimensional complex number system.

One problem can be considered. For example, the vector cross product of two vectors, $A = a_1\mathrm{i} + a_2\mathrm{j} + a_3\mathrm{k}$ and $B = b_1\mathrm{i} + b_2\mathrm{j} + b_3\mathrm{k}$, is defined as

$$A \times B = (a_2\mathrm{b}_3 - a_3b_2)\,\mathrm{i} - (a_2\mathrm{b}_3 - a_3b_2)\,\mathrm{j} + (a_1b_2 - a_2\,b_1)\,\mathrm{k}$$

where i, j, and k are three unit vector for each dimension respectively. The calculation gets complicated and takes much more time especially in the cross product calculation for two three-dimensional vectors.

In terms of time complexity, the time complexity of any scalar number system is less than that of multi-dimensional number system. In order to improve the time complexity for arithmetic operations, there are two possible solutions for solving such a problem as follows:

1. find the new operation system used with multi-dimensional number system in stead of using the scalar number system one, and
2. convert the multi-dimensional number system into a scalar number system form, perform the calculation, and then convert the result into the multi-dimensional number system format.

This thesis focuses on the latter one using the classical computation concept and three-dimensional vectors represented as ai + bj + ck where a, b, and c are real numbers.

The three-dimensional vector has been arised in a number of disciplines in science field including the following domains [3]:

1. Mechanism: Gravatational fields: At each point, the vector gives the direction and magnitude of the force on a particle.
2. Electricity and Magnetism: Electric and magnetic fields, At each point, the vector gives the direction and magnitude of the force on a particle.
3. Fluid Mechanics: Velocity fields: At each point, the vector gives the velocity of a fluid.

In 1977, Ercegovac and Trivedi in [4] had first proposed an important result called *on-line computation* theorem. In fact, in an on-line arithmetic, operands and results flow through arithmetic units in a digit serial manner which is in a most significant digit first mode (MSDF). Since all operations are performed in the same direction, the pipeline concept in which several tasks could be done simultaneously can be applied.

The aim of this thesis is to propose a representation for three-dimensional vector system and also to introduce its on-line fundamental arithmetic operations which are addition, subtraction, and multiplication (cross product). The technique of dimension reduction in order to reduce the calculation effort is also applied. On-line computation is also studied in this work, then a redundant system is combined into the system. The organization of the thesis will be divided into each chapter as follows:

Chapter 2: This chapter introduces the fundamental definitions such as '*number system*' including both signed and unsigned digit number representation systems, '*three- dimensional vector system*', and '*on-line arithmetic*' including the basic algorithms and examples of online addition, subtraction, and multiplication.

Chapter 3: This chapter introduces the new representation of any three-dimensional vector. This new representation will be used for further calculation on on-line arithmetic operations mentioned in Chapter *2* which are addition, subtraction, and multiplication.

Chapter 4: This chapter describes how both on-line addition and subtraction for any three-dimensional vector system can be done. The newly created algorithm will also be introduced. The algorithm proof as well as an example are also stated in this chapter. The multiplication operation on on-line mode will be introduced in this chapter with the newly created algorithm. Same as the on-line addition and subtraction, the proof of the on-line multiplication and example will also be described at the end of this chapter.

Chapter 5: The thesis conclusion will be placed in the last chapter. The reference section will come afterwards.

Objectives

1. To deeply understand the concepts of an on-line arithmetic.
2. To develop a representation for three dimensional vectors and to develop their on-line fundamental arithmetic algorithms, i.e., addition, subtraction and multiplication.

Scope of works

1. To develop a representation for three-dimensional vectors.
2. To develop on-line fundamental arithmetic algorithms, *i.e.*, addition, subtraction, and multiplication (cross product).

Research procedures

1. Study the Knuth complex number system.
2. Study three-dimensional vector system.
3. Study on-line arithmetic algorithms.

4. Develop a new representation for three-dimensional vectors and their on-line fundamental arithmetic algorithms, *i.e.*, addition, subtraction, and multiplication (cross product).
5. Check for newly created algorithms correctness.
6. Make a research conclusion.
7. Complete the research.

Expected results

Three-dimensional vectors in i, j, and k format can be represented by a new representation and can be computable by using their on-line fundamental arithmetic algorithms, *i.e.*, addition, subtraction, and multiplication (cross product).

CHAPTER II

PRELIMINARIES

In this chapter, we recall some definitions and notations that we used in this work. It is started by the representation of number systems. Signed digit number systems of Avizienis are focused our works. Complex number system and three-dimensional vector representation are also recalled. The concept of an on-line computation is also described. Finally, some on-line arithmetic operations are recalled.

2.1 Number Systems

A *number system* can be represented as (β, D) where β is a base and D is a finite digit set of real or complex numbers. The base β can also be either real or complex number. In the classical number system, a canonical digit set, {0, 1, 2, 3, …, β-1}, is used for the base β. Below displays a *β-representation* of X with β as its base,

$$X = (\, X_a X_{a-1} X_{a-2} X_{a-3} X_{a-4} \ldots X_0 . X_{-1} X_{-2} X_{-3} \ldots \,)_\beta,$$

where X_i is an element in D for an integer $i \leq a$. Normally, the set of β-representations on D is denoted by $P[\beta, D]$. The sets of both finite and infinite β-representations are displayed as follows:

- $P_a^b\,[\beta,D]$ denotes the set of all finite β-representations described by

$$\{\ (\, X_a X_{a-1} \ldots X_{b+1} X_b \,)_\beta |\ X_i \in D,\ b \leq i \leq a\ \},$$

- $P_a\,[\beta,D]$ denotes the set of all β-representations described by

$$\{\ (\, X_a X_{a-1} \ldots)_\beta |\ X_i \in D,\ i \leq a\ \},$$

where a and b are the maximum and the minimum degrees respectively.

The *numerical value* of the representation $X = (\, X_a X_{a-1} X_{a-2} \ldots X_b)_\beta$ with base β, denoted by $\| X \|$, can be computed as the following equation:

$$\| X \| = \sum_{i=a}^{b} X_i \beta^i .$$

An important characteristic for a number representation system is to preserve a *lexigographic ordering property*. The definitions of the lexicographic order on β-representations with real numbers of all digits are given as the following definitions.

Definition 2.1

Two β-representations in $P_a[\beta,D]$ which are $X = (X_a X_{a-1} \ldots)_\beta$ and $Y = (Y_a Y_{a-1} \ldots)_\beta$ are said to be comparable. *The representation X is smaller than Y ($X < Y$) in terms of lexicographic ordering if there exists an integer $k \le a$ such that*

$$X_a = Y_a, X_{a-1} = Y_{a-1}, \ldots , X_{k+1} = Y_{k+1}, \text{ and } X_k < Y_k.$$

Definition 2.2

The number system (β,D) has the lexicographic order property *if for any two representations X and Y,*

$$X < Y \text{ if and only if } \| X \| < \| Y \|.$$

Note that, for any two different β-representations X and Y on D, such a system is called to be *redundant* if $\| X \| = \| Y \|$.

2.1.1 Signed digit number system

Redundancy is used extensively for speeding up arithmetic operations. Remarkable examples are signed-digit number systems introduced by Avizienis in 1961, see detail in [5]. These systems are proposed to use some positive and negative integers as digits. This can limit the carry propagation in atithmetic operations. Signed digit number system composed of a finite set of digits and a base. Avizienis proposed to use a digit set $D = \{-1, 0, 1\}$ when the base β is 2. For any integer base $\beta \ge 3$, the set D is represented by a set of the form $\{ e \in \mathbf{Z} \mid -d \le e \le d \}$ where $\beta/2 < d \le \beta-1$. The generalization of signed digit number systems, given by Parhami in 1990 [6], or see detail in [7], can be defined by the following definition.

Definition 2.3

The signed digit number system (β, D) is composed of a base β where β is a positive integer ≥ 2 and a digit set D,

$$D = \{ e \in Z \mid a \leq e \leq b \},$$

where a and b are integers such that $a \leq 0 \leq b$.

Remark 2.1

1. *Negative numbers cannot be represented in this system if $a = 0$.*
2. *Positive numbers cannot be represented in this system if $b = 0$.*

The number of digits in a digit set D is equal to $| D | = b - a + 1$. This number can describe the *redundancy property* of the number representation system. The system is said to be a *redundancy* system if there is at least one number can have more than one representation.

Remark 2.2

1. *If $| D | < \beta$, some reals cannot be represented in the system.*
2. *If $| D | = \beta$, every integer has a finite representation, and every real number can be represented.*
3. *If $| D | > \beta$, this system is* redundant.

Definition 2.4

1. *The digit set D is a* minimally redundant *digit set if $| D | = \beta + 1$.*
2. *The digit set D is a* maximally redundant *digit set if $| D | = 2\beta - 1$.*
3. *The digit set D* sysmetric *if $b = | a |$.*

For instance, in the base $\beta = 2$ with a digit set $D = \{ -1, 0, 1 \}$, number 14 can have more than one representation as illustrated by Fig.2.1. That is 14 can be written as $(001110)_2$ or $(0100\text{-}10)_2$.

β^5	β^4	β^3	β^2	β^1	β^0
32	16	8	4	2	1
0	0	1	1	1	0
0	1	0	0	-1	0

Figure. 2.1 The representation of 14 in the number system (2, {-1, 0, 1})

2.1.2 Negative integer base number system

Signed digit number system can be extented using a negative integer as a base. Let β be the base with β < -1. A real number is represented in any negative base β containing digits in $D = \{ e \in Z \mid 0 \leq e \leq |\beta| - 1 \}$. Remark 2.2 is also applied with this number system.

2.1.3 Complex number

A complex number sometimes can be treated as a two-dimensional vector system where the number of the real number part can be matched to the first dimension part whereas the number of the imaginary part can also be matched to the second dimension part on the *Cartesian coordinate system.*

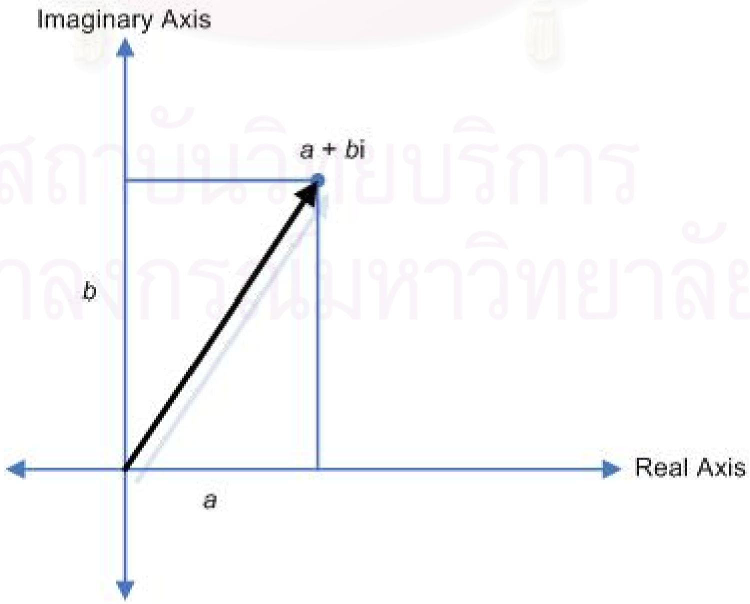

Figure 2.2 Complex number representation on two-dimensional plane

Definition 2.5

The complex number system can be defined as ordered pairs of real numbers (a, b).

Addition: *(a , b) + (c , d) = (a + c, b + d)*

Multiplication: *(a , b) + (c , d)* **=** *(ac – bd, bc + ad)*

Definition 2.6

Given a complex number A = (a, b), the size *(or length) of A, denoted by || A ||, is defined as*

$$\| A \| = \sqrt{a^2 + b^2} \ .$$

In fact, a vector was firstly discovered in the first two decades of 19^{th} century with the geometric representations of complex numbers. Caspar Wessel (1745-1818), Jean Robert Argand (1768-1822), and Carl Friedrich Gauss (1777-1855) conceived that complex number can be as points in the two-dimesional plane, eg. two-dimensional vectors [8].

A vector is basically a specific mathematical structure. It has numerous physical and geometric applications, which result mainly from its ability to represent magnitude and direction simultaneously. The location of points on a cartesian coordinate plane is usually expressed as an ordered pair (*x*, *y*), which is a specific example of vector. A vector (*x*, *y*) has a certain distance and angle relatively from the origin (0,0). In general, vector can be described as a multi-dimensional representation.

For these reasons, fundamental arithmetic operations for two-dimensional vectors can be performed in the complex number system.

Example 2.1

Find the addition of two two-dimensional vectors 3i + 4j and 4i + 9j

Solution

This can be solved by adding two complex numbers, (3, 4) and (4, 9). Therefore, the result will come up with (3+4) i + (4+9) j = 7i + 13j.

Engineering computation in vectors often uses both dot product and cross product. The dot product gives the vector amount that one vector contributes along the same line to another vector. The cross product, however, is partly the result of multiplying different components of two vectors to get a product vector that is lying perpendicularly to both of the original vectors.

Definition 2.7

Given two two-dimensional vectors, $A = (a, b)$ and $B = (c, d)$, a dot product *of A and B, denoted by $A{\cdot}B$, can be computed as*

$$A{\cdot}B = \|A\| \times \|B\| \times \cos\theta$$

where θ is an angle between vector A and B.

Definition 2.8

Given two two-dimensional vectors, $A = (a, b)$ and $B = (c, d)$, a cross product *of A and B, denoted by $A{\times}B$, can be computed as*

$$A{\times}B = (ac - bd, ad + bc).$$

2.1.4 Three-dimensional vector system

Unlike any two-dimensional vector system having an ordered pair (*a*, *b*), three-dimensional vector is an ordered triplet (*a*, *b*, *c*) where *a*, *b*, and *c* are any reals. In fact, points in a plane or in three-dimensional space can be considered as vectors. The representation of vector (*a*, *b*, *c*) in a three-dimensional space is illustrated by Fig.2.3. The higher dimensional vector is, the more useful the vector can be used. The definition of three-dimensional vector can be described as follows:

Definition 2.9

Let a, b, and c be three reals. The vector representation of the three-dimensional vector (a ,b, c) can be expressed as a summation of products of the unit vectors i, j, and k with a, b, and c respectively. Then the representation of the vector V is written as

$$V = ai + bj + ck.$$

Definition 2.10

Given a three-dimensional vector A = (a, b, c) where a, b, and c are reals. The additive inverse *of vector A, usually denoted by* A^{-1}*, is a vector of the form (-a, -b, -c).*

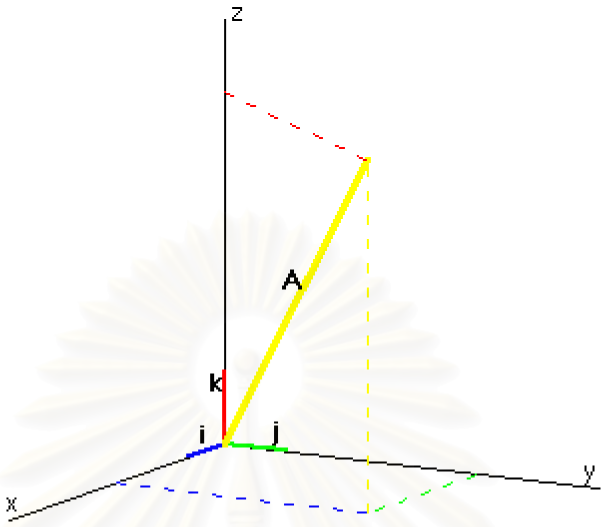

Figure 2.3 The representation of three-dimensional vector system

Supposingly, given two three-dimensional vectors which are $V_1 = ai + bj + ck$ and $V_2 = di + ej + fk$ where a, b, c, d, e, and f are real numbers. Addition of vectors is to construct a path of a sequence of all added vectors started from the origin (0, 0, 0). Addition in this system are defined as addition for each dimension seperately. This means that addition of the two vectors is a vector

$$V_1 + V_2 = (a + d)\text{ i} + (b + e)\text{ j} + (c + f)\text{ k}.$$

Subtraction can be considered as an addition of its additive inverse. For instance, let V_1 be expressed by 2i + 3j + 4k and let V_2 be expressed by 4i + 3j + 5k. Addition (subtraction) of the both vectors can be done as follows:

$$V_1 + V_2 = (2+4)\text{ i} + (3+3)\text{ j} + (4+5)\text{ k} = 6\text{i} + 6\text{j} + 9\text{k}$$

$$V_1 + V_2 = (2-4)\text{ i} + (3-3)\text{ j} + (4-5)\text{ k} = -2\text{i} - \text{k}$$

It is clear that it would take $\Theta(n)$ for both addition and subtraction where n is the number of digits used to represent the two vectors (*i.e.*, the number of dimensions).

For multiplication operation in this system, it is defined as a *cross product* which applies the concept of *right hand finger* shown as Fig.2.4.

Definition 2.11

Given two three-dimensional vectors $A = (a, b, c)$ and $B = (d, e, f)$ where $a, b, c, d, e,$ and f are reals. A cross product of A and B is defined as

$$\begin{aligned} A \times B &= (ai \times di) + (ai \times ej) + (ai \times fk) \\ &+ (bj \times di) + (bj \times ej) + (bj \times fk) \\ &+ (ck \times di) + (ck \times ej) + (ck \times fk). \end{aligned}$$

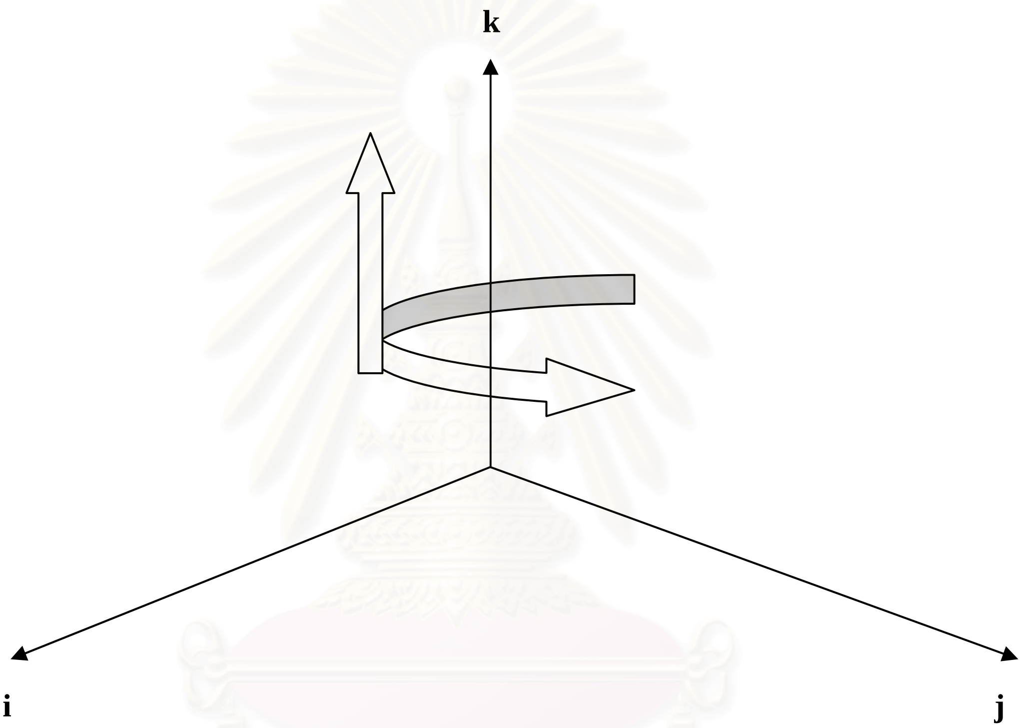

Figure 2.4 Cross product calculation of three-dimensional vector system based on righthand finger concept

The cross product of unit vectors can be described by the following definition.

Definition 2.12

Let i, j, and k be three unit vectors. The cross product of two unit vectors can be described as

$$\begin{array}{lll} i \times i = 0, & i \times j = k, & i \times k = -j, \\ j \times i = -k, & j \times j = 0, & j \times k = i, \\ k \times i = j, & k \times j = -i, & k \times k = 0. \end{array}$$

Then the cross product of any two vectors can be considered as nine multiplication operations of two scalar numbers and another eight addition operations of them. The complexity is obviously equal to $O(n^2)$.

Example

let $V_1 = 2i + 3j + k$ and let $V_2 = i + 2j + 2k$. Find the cross product of V_1 and V_2.

Solution

The cross product of V_1 and V_2 can be computed as follows.

$$V_1 \times V_2 = 2{\times}1\,[\,i \times i\,] + 2{\times}2\,[\,i \times j\,] + 2{\times}2\,[\,i \times k\,] + 3{\times}1\,[\,j \times i\,] + 3{\times}2\,[\,j \times j\,] + 3{\times}2\,[\,j \times k\,] + 1{\times}1\,[\,k \times i\,] + 1{\times}2\,[\,k \times j\,] + 1{\times}2\,[\,k \times k\,].$$

According to the concept of right hand finger illustrated above, the solution of this operation can be broken down as follows:

COLUMN		**i**	**j**	**k**
$V_1 \times V_2$	=	$(3{\times}2 + (-1{\times}2))$	$(-2{\times}2) + (1{\times}1)$	$(2{\times}2 + (-3{\times}1))$
	=	4i	(-3j)	1k
	=	4i – 3j + k.		

This takes a significance of time to solve the cross product between both three dimensional vectors.

2.2 On-line Arithmetic

Most fundamental operations for any base number system, say, base 10 (decimal number system) such as multiplication, addition, and subtraction are normally done by using the least significant digit first (LSDF) mode which means the steps of operations are computed from the *rightmost* digit, kept moving left until all done. But division is performed in the most significant digit first (MSDF) mode. In order to pipeline the operations (*i.e.*, each operation can be started without waiting for the end of the previous operation), every operation should be processed in the same direction. For that purpose, on-line arithmetic was first introduced by Ercegovac and Trivedi [11]. The operands and the results flow serially through arithmetic units, digit-by-digit, starting from the most significant digit. On-line systems can also be characterized by the on-line delay δ, the smallest integer which is the first n digits of the result can be deduced from the first $n + \delta$ digits of the inputs. In order to do this, the number representation system should be a redundant system. The signed digit number system is selected in the on-line arithmetic computation system.

2.2.1 On-line addition

The following theoretical result shows that an on-line addtion can be perform with the on-line delay δ using a redundant number system.

Theorem 2.1

Let β be an integer, $\beta > 1$ and let D be a finite set, $D = \{ -b, -b\text{-}1, \ldots, 0, 1, \ldots, b \}$ where b is an integer such that $\beta/2 \le b \le \beta - 1$. On-line addition can be computable in the system (β, D) with an on-line delay δ, where

$$\delta = \begin{cases} 2 & : b = \beta/2 \\ 1 & : otherwise \end{cases}.$$

The proof of the theorem is to proposed an on-line addition algorithm shown as below, (the detail of the proof, the reader can see in [9].

Algorithm: OnlineAddition

Input: $X := (x_m x_{m-1} \ldots)_\beta$ and $Y = (y_m y_{m-1} \ldots)_\beta$ where $x_i, y_i \in D$

Output: $Z := (z_m z_{m-1} \ldots)_\beta$ where $z_i \in D$

begin

$r_m := \sum_{i=m-\delta+1}^{m} (x_i + y_i)\beta^{i-m+\delta-1}$;

$j := m$;

while $j \le m$ **do**

s_j := (x_j + y_j);

r_{j-1} := (x_j + y_j) + $r_j\beta$ – ($\beta^\delta \times \lfloor (x_j + y_j + r_m\beta) / \beta^\delta \rfloor$);

z_j := (x_j + y_j + $r_j\beta$ - r_j) / β^δ;

j := j – 1;

enddo;

end;

Example 2.1

In the system with base β = 5 with digits in D = {-3, -2, -1, 0, 1, 2, 3},on-line addition of 708 and 766 with an on-line delay δ = 1 can be expressed as Fig. 2.5.

Solution

		β^5	β^4	β^3	β^2	β^1	β^0	
708	=		1	1	-2	1	3	
								+
766	=		1	1	1	-2	1	
			2	2	-1	-1	4	
Remainder			2	2	-1	-1	-1	
Carry digit		0	0	0	0	1		
		0	2	2	-1	0	-1	

Figure. 2.5 The addition of 708 and 766 in the system (5, {-3, -2, -1, 0, 1, 2, 3 })

The solution is (0 2 2 -1 0 -1 $)_5$ = $(2\times5^4) + (2\times5^3) + (-1\times5^2) + (-1)$

= 1250 + 250 – 25 -1 = 1474.

It is clear that the result can be produced digit-by-digit , starting from the most significant digit, with the on-line delay 1. For instance, -1 at the position β^1 cannot be outputed until the carry propagation of the column β^0 is known. In this system, the effect of the carry propagation is limitted to only one digit on the left.

2.2.2 On-line subtraction

On-line subtraction would take a concept of what the on-line addition is. The on-line addition algorithm is still used for achieving the subtraction task. For instance, an on-line subtraction of B from A can be considered as an on-line addition of A and –B.

Example

In base β = 5 with digits in D = {-3, -2, -1, 0, 1, 2, 3}, subtraction 766 from 708 can be computed as follows:

Solution

The same calculation procedures as the example above would apply for this subtraction calculation as well. The only difference is that to convert number 766 which is the subtracter to be a negative value in stead of the positive one, as shown in Fig. 2.6.

The solution is $(0\ \ 0\ \ -1\ \ 3\ \ -2\ \ 2)_5 = (-1\times5^3) + (3\times5^2) + (-2\times5) + (2)$

$= -125 + 75 - 10 + 2 = -58.$

	β^5	β^4	β^3	β^2	β^1	β^0	
708	=	1	1	-2	1	3	
							+
-766	=	-1	-1	-1	2	-1	
		0	0	-3	3	2	
Remainder		0	0	2	-2	2	
Carry digit	0	0	-1	1	0		
	0	**0**	**-1**	**3**	**-2**	**2**	

Figure. 2.6 The subtraction of 766 from 708 in the system (5, {-3, -2, -1, 0, 1, 2, 3 })

2.2.3 On-line multiplication

On-line multiplication uses the combination of both incremental multiplication technique and redundant number system. The on-line multiplication has a delay δ depending on the range of the input. This delay can be ignored by adding δ zeroes at the left part of each operand. Then each operand is less than $1/\beta^{\delta}$ called *the operand bound*. The generic on-line algorithms for real and complex representation are studied in [10].

The classical on-line multiplication algorithm illustrated below needs to be known for fundamental understading of the on-line multiplication concept. The on-line multiplication can be described by the following theorem.

Theorem 2.2

Let β be an integer, β > 1 and let D be a finite set, D = { -b, -b-1, …, 0, 1, …, b } where b is an integer such that β/2 ≤ b ≤ β – 1. On-line multiplication can be computable in the system (β, D) with an on-line delay δ, where

$$\delta = \begin{cases} 1 : \beta > 3 \\ 2 : \beta = 2,3 \end{cases}.$$

The on-line multilication algorithm is as follows (the proof of the algorithm can be found in [2]):

Algorithm: OnlineMultiplication

Input: $A = (.a_{-1}a_{-2}\ldots)_{\beta}$ and $B = (.b_{-1}b_{-2}\ldots)_{\beta}$

Output: $X = (.x_{-1}x_{-2}\ldots)_{\beta}$ where $\| X \| = \sum_{j=-1}^{\infty} x_j \beta^{j} = \| A \| \times \| B \|$

begin

$x_{-1} := x_{-2} := \ldots := x_{-\delta} := 0;$

$W_{-\delta} := 0;$

$j := -\delta-1;$

while $j \leq -\delta-1$ **do**

$W_j := \beta\,(W_{j+1} - x_{j+1}) + A_j b_j + B_{j+1} a_j$;

if $|W_j| \leq b$ **then** $x_j :=$ Sign$(W_j) \lfloor |W_j| + 1/2 \rfloor$

else $x_j :=$ Sign$(W_j) \lfloor |W_j| \rfloor$ **endif;**

$j := j\text{-}1$;

enddo;

end;

Example

Let A and B be two numbers in base 2 with digits in D = { -1, 0, 1 },

where A=B = $(.0011011111)_2$

Solution

Table 2.1 displays the result of a multiplication of *A* and *B*.

Table 2.1 On-line multiplication of *A* and *B* where $A = B = (.0011011111)_2$

j	$A_j b_j + B_{j+1} a_j$	W_j	x_j	$2(W_j - x_j)$
-1	0.0	0.0	**0**	0.0
-2	0.0	0.0	**0**	0.0
-3	0.001	0.001	**0**	0.01
-4	0.0101	0.1001	**1**	-0.111
-5	0.0	-0.111	**-1**	0.01
-6	0.011001	0.101001	**1**	-0.10111
-7	0.0110101	-0.0100111	**0**	-0.100111
-8	0.01101101	-0.00101111	**0**	-0.0101111
-9	0.011011101	0.000100001	**0**	0.00100001

The output after computation would be $(.0001\text{-}11000\ldots)_2$.

CHAPTER III

THREE-DIMENSIONAL VECTOR REPRESENTATION

This chapter introduces the newly created representation of any three-dimensional vector system. This novel representation is proposed in order to simplify the computation of three-dimensional vectors. Some fundamental vector operations (*i.e.*, addition, subtraction and cross product) are also introduced in this chapter.

3.1 Introduction

Normally, any complex number system is used to express a two-dimensional vector system. For instance, a two-dimensional vector (x, y) can be considered as the pair of x and y which are popularly used to represent the real part of the number and its imaginary part respectively. Using this concept, a two-dimensional vector (x, y) is expressed by a complex number of the form $x + y\mathrm{i}$.

Any vector can be described as a complex number system; therefore, the concept of an above complex number system representation can be extended to describe any three-dimensional number system as well.

It is not convenient to maintain a complex number using two dependent parts (real and imaginary). One can be applied to combine the both parts together, that is to use a complex base representation with some integer digit sets, for instance Knuth's complex number representation systems [11]. On the other hand, any complex number can also be represented in the real base number system with a finite set of complex digits. The later is focused in our work.

In order to perform the computation in an on-line mode, the concept of redundant number system on which such a number can have more than one representation is also applied. Especially, Avizienis's signed digit number representation [5] is interested in this research.

3.2 The representation

Classical representation of a three-dimensional vector (a, b, c) is written as

$$a\text{i} + b\text{j} + c\text{k},$$

for any real numbers a, b, and c. the variables i, j, and k are represented unit vectors of each dimension respectively. Now we will propose a novel three-dimensional vector representation system. A vector is represented by a sequence of signed-vector-digit. We also show that all three-dimensional vectors can have a representation in this novel system.

Definition 3.1

Let β be a positive integer, $\beta \geq 2$, and let D be a finite set of three-dimensional vectors of the form

$$\{(x, y, z) \mid -b \leq x, y, z \leq b\}$$

where b is an integer such that $\beta/2 \leq b \leq \beta-1$. The signed-vector-digit representation of a three-dimensional vector X is written as:

$$X_s\, X_{s-1}\, X_{s-2} \ldots X_t$$

where $X_i \in D$ for all i, $t \leq i \leq s$.

For instance, in base $\beta = 2$ with a digit set $D = \{ (x, y, z) \mid -1 \leq x, y, z \leq 1 \}$, vector (12, -3, 5) can be represented in this system as

$$(1, 0, 0)\ (1, -1, 1)\ (0, 0, 0)\ (0, 1, 1).$$

The value of this representation can be computed as follows:

$$(1, 0, 0)2^3 + (1, -1, 1)\, 2^2 + (0, 0, 0)\, 2^1 + (0, 1, 1)\, 2^0 = (12, -3, 5).$$

Since the proposed system is redundant, vector (12, -3, 5) can also be represented by the other sequence as

$$(1, 0, 0)\ (0, -1, 0)\ (-1, 1, 1)\ (0, 0, 1)\ (0, 1, -1).$$

Definition 3.2

Let β be a real number, $\beta \geq 2$. Given a signed-vector-digit representation

$$X = X_s\, X_{s-1}\, X_{s-2} \ldots X_t$$

in the base β where $X_i \in \{ (x, y, z) \mid -b \le x, y, z \le b \}$, $\beta/2 \le b \le \beta - 1$, for any integer i, $t \le i \le s$. The complex value of X, denoted by $\| X \|$ is

$$\|X\| = X_s\beta^{s} + X_{s-1}\beta^{s-1} + X_{s-2}\beta^{s-2} + \ldots + X_t\beta^{t}.$$

Let a three-dimensional vector $X = X_s\, X_{s-1}\, X_{s-2} \ldots X_t$, and $X_i = (x_{i1}, x_{i2}, x_{i3})$. That is

$$X_i = x_{i1}\mathrm{i} + x_{i2}\mathrm{j} + x_{i3}\mathrm{k}.$$

The complex number that is matched to X is

$$\begin{aligned} X &= (x_{s1}\mathrm{i} + x_{s2}\mathrm{j} + x_{s3}\mathrm{k})\beta^{s} + (x_{(s-1)1}\mathrm{i} + x_{(s-1)2}\mathrm{j} + x_{(s-1)3}\mathrm{k})\beta^{s-1} \\ &\quad + X_{s-2}\beta^{s-2} + \ldots + X_t\beta^{t} \\ &= \sum_{u=t}^{s} x_{u1}\beta^{u}\,\mathrm{i} + \sum_{u=t}^{s} x_{u2}\beta^{u}\,\mathrm{j} + \sum_{u=t}^{s} x_{u3}\beta^{u}\,\mathrm{k} \end{aligned}$$

Example 3.1

In the base n = 2, with a digit set $D = \{(x,y,z) \mid -1 \le x, y, z \le 1 \}$,given a signed-vector-digit representation X = ((1, 0, 1) (1, 1, 0) (0, 1, 1)), find the complex value of X.

Solution

$$\begin{aligned} X &= (1\mathrm{i} + 0\mathrm{j} + 1\mathrm{k})2^{2} + (1\mathrm{i} + 1\mathrm{j} + 0\mathrm{k})2^{1} + (0\mathrm{i} + 1\mathrm{j} + 1\mathrm{k})2^{0} \\ &= (4\mathrm{i} + 4\mathrm{k}) + (2\mathrm{i} + 2\mathrm{j}) + (\mathrm{j} + \mathrm{k}) \\ &= 6\mathrm{i} + 3\mathrm{j} + 5\mathrm{k}. \end{aligned}$$

3.3 Completeness

The proposed vector representation system uses a real number as the base and a complex numbers as digits in the system. Now we will show that any three-dimensional vector can have a representation in the proposed system. This is shown in the following theorem.

Theorem 3.1

Given a three-dimensional vector $V = (u, v, w)$ where u, v, and w are any real numbers, the vector V can have a representation in the signed-vector-digit representation system in base β with digits in $D =\{(x, y, z) \mid -b \le x, y, z \le b\}$ where b is an integer such that $\beta/2 \le b \le \beta-1$.

Proof:

It is clear that the three-dimensional vector $V = (u, v, w)$ can be expressed by a complex number of the form ui + vj + wk. Since u, v, and w are all real numbers, they have a representation in base β with digits in E $\{ e \mid -b \le e \le b \}$. Let

$$u = (u_t u_{t-1} u_{t-2\ldots}\ u_s)_\beta \quad \text{where } u = \sum_{j=t}^{s} u_j \beta^j$$

$$v = (v_t v_{t-1} v_{t-2\ldots}\ v_s)_\beta \quad \text{where } v = \sum_{j=t}^{s} v_j \beta^j$$

$$w = (w_t w_{t-1} w_{t-2\ldots}\ w_s)_\beta \quad \text{where } w = \sum_{j=t}^{s} w_j \beta^j$$

This is of course,

$$\begin{aligned} V &= u\text{i} + v\text{j} + w\text{k} \\ &= \sum_{j=t}^{s} u_j \beta^j\ \text{i} + \sum_{j=t}^{s} v_j \beta^j\ \text{j} + \sum_{j=t}^{s} w_j \beta^j\ \text{k} \\ &= (u_t + v_t + w_t)\beta^t + (u_{t-1} + v_{t-1} + w_{t-1})\beta^{t-1} + \ldots + (u_s + v_s + w_s)\beta^s \\ &= (u_t, v_t, w_t)\ (u_{t-1}, v_{t-1}, w_{t-1})\ \ldots\ (u_s, v_s, w_s) \text{ where } u_i, v_i, \text{ and } w_i \text{ are in } E. \end{aligned}$$

The proof is completed.

Example 3.2

Given a vector $V = 6i + 13j - 5k$, find the signed-vector-digit representation of V in base 2 with digits in $D = \{ (x, y, z) \mid -1 \le x, y, z \le 1 \}$.

Solution

Since $6 = (110)_2$, $13 = (1101)_2$, and $-5 = (-10-1)_2$, then the representation of V is

$$(0, 1, 0)\ (1, 1, -1)\ (1, 0, 0)\ (0, 1, -1).$$

3.4 Redundancy

Any three-dimensional vector system can have more than one signed-vector-digit representation. Therefore, a three-dimensional vector system is *redundant.* This gives the flexibility to have various signed-vector-digit representations to the system to any particular three-dimensional vector.

Theorem 3.2

In the vector representation with base β and a digit set $D = \{(x, y, z) \mid -b \le x, y, z \le b\}$ where b is an integer such that $\beta/2 \le b \le \beta-1$, any three-dimensional vector can have more than one finite representation.

Proof:

Let V be a three-dimensional vector, $V = (u, v, w)$. Since the number representation system in base β with digits in $E = \{ e \mid -b \le e \le b \}$ is a redundant number system, u, v, and w must have more than one representation in (β, E). Then,

$$u = (u_t u_{t-1} u_{t-2\ldots}\, u_s)_\beta \quad \text{where } u = \sum_{j=t}^{s} u_j \beta^j ,$$

$$v = (v_t v_{t-1} v_{t-2\ldots}\, v_s)_\beta \quad \text{where } v = \sum_{j=t}^{s} v_j \beta^j ,$$

$$w = (w_t w_{t-1} w_{t-2\ldots}\, w_s)_\beta \quad \text{where } w = \sum_{j=t}^{s} w_j \beta^j .$$

It is also true that there exists other representations for u, v, and w, then

$$u = (d_t d_{t-1} d_{t-2\ldots}\, d_s)_\beta \quad \text{where } d = \sum_{j=t}^{s} d_j \beta^j ,$$

$$v = (g_t g_{t-1} g_{t-2\ldots}\, g_s)_\beta \quad \text{where } g = \sum_{j=t}^{s} g_j \beta^j ,$$

$$w = (h_t h_{t-1} h_{t-2\ldots}\, h_s)_\beta \quad \text{where } h = \sum_{j=t}^{s} h_j \beta^j .$$

This is of course,

$$V = u\mathrm{i} + v\mathrm{j} + w\mathrm{k} = d\mathrm{i} + g\mathrm{j} + h\mathrm{k}.$$

This can imply that V has more than one representation in base β with digits in D.

Example 3.3

In the base β = 2, with a digit set D = {(x, y, z) | - 1 ≤ x, y, z ≤ 1 }, the vector V_1 *=* *4i + 3j + 5k can be represented as*

$$V_1 = ((1, 0, 1)\ (0, 1, 0)\ (0, 1, 1))_2.$$

Find one different representation of V_1*.*

Solution

This number system is also a redundant system meaning that there is more than one representation for denoting this vector. As a result of that, V_1 can also be written as

$$V_1 = ((1, 1, 0)\ (-1, -1, 1)\ (0, 0, 1)\ (0, -1, -1))_2.$$

Example 3.3

In the base number β = 3 with the digit set D = {(x, y, z) | - 2 ≤ x, y, z ≤ 2}, the vector $V_2 = 9i + 8j + 12k$ *can be represented as*

$$V_2 = ((1, 0, 1)\ (0, 2, 1)\ (0, 2, 0))_3.$$

Find a different representation of V_2*.*

Solution

With the redundant number system concept, one of other representations of V_2 can also be described as follows:

$$V_2 = ((1, 1, 1)(-2, -2, -1)\ (0, 0, -2)\ (0, -1, 0))_3$$

Then, the proposed vector representation system is completed and also has a redundant property. The next chapter will show how to perform an on-line computation in this system.

CHAPTER IV

ON-LINE ARITHMETIC OPERATIONS FOR THREE-DIMENSIONAL VECTOR REPRESENTATION

This chapter will introduce the fundamental arithmetic operations which are addition, subtraction, and multiplication for any two three-dimensional vectors in an on-line mode with the signed-vector-digit representation. The newly created algorithm and the proof for those arithmetic operations will be introduced. At the end of the chapter, the examples of them will be shown.

The on-line arithmetic operations for three-dimentsional vector system brings the concept of the on-line arithmetic operation to apply which is the most significant digit first calculation mode. That means all operation would be done in the same direction for the leftmost digit to the rightmost digit while the result for each dimension would be produced along the way without waiting until the end of the precedent operation.

4.1 On-line addition

The algorithm below illustrates the on-line addition algorithm for a signed-vector-digit representation system. In order to avoid the overflow problem, let us assume that the first signed-vector-digit of each operand is (0, 0, 0).

Theorem 4.1

Given two three-dimensional vectors $V = (v_0, v_1, v_2)$ and $W = (w_0, w_1, w_2)$ where v_0, v_1, v_2, w_0, w_1, and w_2 are real numbers. Let the signed-vector-digit representation in base β with digits in D of V and W be

$$V = ((v_{0,t}, v_{1,t}, v_{2,t})(v_{0,t-1}, v_{1,t-1}, v_{2,t-1})(v_{0,t-2}, v_{1,t-2}, v_{2,t-2})\ldots(v_{0,s}, v_{1,s}, v_{2,s}))_\beta$$

and

$$W = ((w_{0,t}, w_{1,t}, w_{2,t})(w_{0,t-1}, w_{1,t-1}, w_{2,t-1})(w_{0,t-2}, w_{1,t-2}, w_{2,t-2})\ldots(w_{0,s}, w_{1,s}, w_{2,s}))_\beta,$$

where $D = \{(x, y, z) \mid -b \le x, y, z \le b\}$ and b is an integer such that $\beta/2 \le b \le \beta-1$. The on-line addition of V and W, denoted by Z, can be performed with an on-line delay δ,

$$\delta = \begin{cases} 2 \ : b = \beta/2 \\ 1 : otherwise \end{cases}.$$

Proof

The proof of the theorem is given by introducing the following on-line addition algorithm.

Algorithm: A_V

Input:

$V=((v_{0,t},v_{1,t},v_{2,t})(v_{0,t-1},v_{1,t-1},v_{2,t-1})(v_{0,t-2},v_{1,t-2},v_{2,t-2})\ldots(v_{0,s},v_{1,s},v_{2,s}))_\beta$

$W=((w_{0,t},w_{1,t},w_{2,t})(w_{0,t-1},w_{1,t-1},w_{2,t-1})(w_{0,t-2},w_{1,t-2},w_{2,t-2})\ldots(w_{0,s},w_{1,s},w_{2,s}))_\beta$

Output:

$Z=((z_{0,t},z_{1,t},z_{2,t})(z_{0,t-1},z_{1,t-1},z_{2,t-1})(z_{0,t-2},z_{1,t-2},z_{2,t-2})\ldots(z_{0,s},z_{1,s},z_{2,s}))_\beta$

begin

$p := t$;

$j := 0$;

while $j \le 2$ **do**

$r_{j,p-\delta} := \sum_{k=t}^{t-\delta+1}(v_{j,k} + w_{j,k})\beta^{k-t+\delta-1}$;

$j := j + 1$;

enddo

while $p \ge s + \delta$ **do**

$j := 0$;

while $j \le 2$ **do**

$s_{j,p-\delta} := v_{j,p-\delta} + w_{j,p-\delta}$;

if $-2b \le s_{j,p-\delta} < -b+1$

then $c_{j,p-\delta+1} := -1$;

$s_{j,p-\delta} := s_{j,p-\delta} + \beta$; **endif**

if $-b+1 \le s_{j,p-\delta} \le b-1$

then $c_{j,p-\delta+1} := 0$; **endif**

if $b-1 < s_{j,p-\delta} \le 2b$

then $c_{j,p-\delta+1} := 1;$

$s_{j,p-\delta} := s_{j,p-\delta} - \beta$; **endif**

$z_{j,p}$ **:=** Integer($(r_{j,p-\delta} + c_{j,p-\delta}) / \beta^{\delta-1}$) ;

$r_{j,p-\delta-1} := ((r_{j,p-\delta} - (z_{j,p} \times \beta^{\delta-1})) \times \beta) + s_{j,p-\delta}$;

$j := j + 1;$

enddo;

$p := p - 1;$

enddo;

$j := 0;$

while $j \le 2$ **do**

If $\delta = 2$ **then Rewrite** $r_{j,s+\delta-1} := z_{s+\delta-1}\beta^{\delta-1} + z_{s+\delta-2}\beta^{\delta-2}$; **endif**

If $\delta = 1$ **then Rewrite** $r_{j,s+\delta-1} := z_{s+\delta-1}\beta^{\delta-1}$; **endif**

$j := j + 1;$

enddo

end;

Where a = Integer (b) means that a is the closest integer to b, for any real number b.

Proof of the algorithm:

In order to prove that the above algorithm is correct, we have to show that the algorithm is valid and correct. Moreover, the complexity of the algorithm will be shown as well.

Validation:

It is to show that the result of the addition operation of two three-dimensional vectors is always in the digit set D as its operands (*i.e.*, the answer is valid in the representation).

It is obvious that addition in each dimension can be computed separately. Then the proof will be done for only one dimension called j. The problem can be transformed into a problem for adding in base β with digits in $E = \{ e \mid -b \le e \le b \}$ of

$V_j = (v_{j,t}, v_{j,t-1}, v_{j,t-2}, v_{j,t-3}, \ldots, v_{j,s})_\beta$ and

$W_j = (w_{j,t}, w_{j,t-1}, w_{j,t-2}, w_{j,t-3}, \ldots, w_{j,s})_\beta$.

The delay for this addition becomes an on-line delay in the signed-vector-digit representation system.

From the algorithm, at the p^{th} iteration,

$r_{j,p-\delta}$ denotes the remainder from the previous iteration,

$s_{j,p-\delta}$ denotes the interim sum at the current iteration,

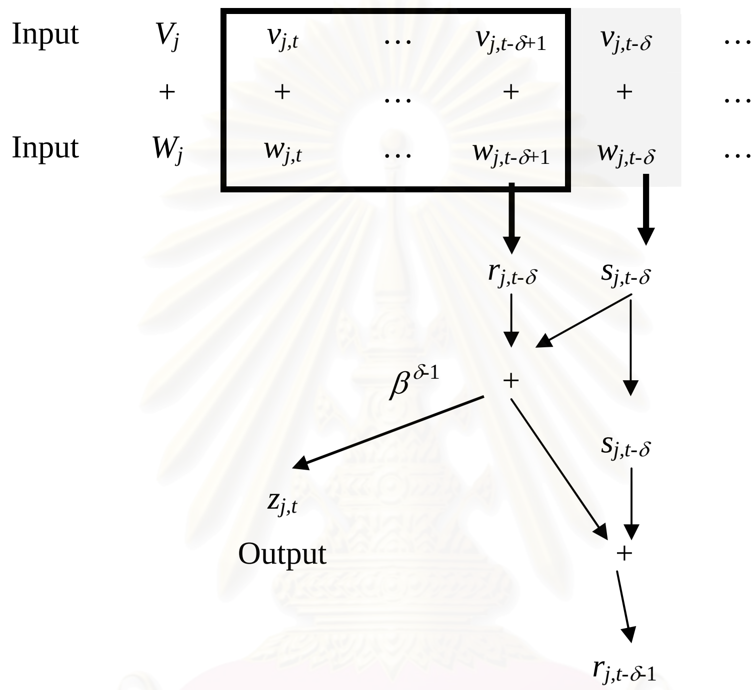

Figure. 4.1 The relation of the inputs, $r_{j,t-\delta}$, $s_{j,t-\delta}$, and the output ($z_{j,t}$)

$c_{j,p-\delta}$ denotes the carry propagation at the current iteration.

The relation at the first iteration can be illustrated by Fig 4.1.

It is remarked that for the first iteration, the remainder $r_{j,p-\delta}$ is computed from the δ first digits of the input operands.

The rest is to show that all $z_{j,p}$, for all $s \le p \le t$, is an element in E. The proof is separated into three cases,

Case 1: $p = t$ (the first iteration)

From the definition of $r_{j,p-\delta}$, $v_{j,t}$ =0 and $w_{j,t}$ = 0, it is obtained that

$$| r_{j,p-\delta}| \le 2b \left(\frac{\beta^{\delta-1} - 1}{\beta - 1}\right). \qquad (4.1)$$

Since $z_{j,p}$ **:=** Integer($(r_{j,p-\delta} + c_{j,p-\delta}) / \beta^{\delta-1}$) and (4.1), then

$$| z_{j,p} | \le \text{Integer}\left(\frac{2b\left(\frac{\beta^{\delta-1}-1}{\beta-1}\right)+1}{\beta^{\delta-1}}\right). \tag{4.2}$$

From the definition, $\beta/2 \le b \le \beta$ -1, then

$$\frac{2b}{\beta-1} \le 2, \text{ and} \tag{4.3}$$

$$\frac{1}{\beta^{\delta-1}}\left(\frac{2b}{\beta-1}-1\right) > 0. \tag{4.4}$$

From (4.2), (4.3) and (4.4), we can conclude that

$$| z_{j,p} | < 2.$$

Case 2: $s+\delta-1 < p < t$

From the algorithm, $z_{j,p}$ **:=** Integer($(r_{j,p-\delta} + c_{j,p-\delta}) / \beta^{\delta-1}$), then

$$r_{j,p-\delta} - (z_{j,p} \times \beta^{\delta-1}) \le \left\lfloor \frac{\beta^{\delta-1}}{2} \right\rfloor. \tag{4.5}$$

From $r_{j,p-\delta-1}$:= (($r_{j,p-\delta}$ - ($z_{j,p} \times \beta^{\delta-1}$)) $\times \beta$) + $s_{j,p-\delta}$, and (4.5), we conclude that

$$| r_{j,p-\delta}| \le \left\lfloor \frac{\beta^{\delta-1}}{2} \right\rfloor \times \beta + b - 1. \tag{4.6}$$

The proof is separated into two cases,

Case 2.1: $\delta = 1$

From (4.6) and , it is obtained that $| r_{j,p-\delta}| \le b - 1$. It is thus

$$| z_{j,p} | < b - 1.$$

Case 2.2: $\delta = 2$

In this case, it is $b = \frac{\beta}{2}$. From (4,6), then $| r_{j,p-\delta}| \le b \times \beta + b - 1$. That is

$$| z_{j,p} | \le \text{Integer}\left(b + \left(\frac{b}{\beta} - \frac{1}{\beta}\right)\right) = b.$$

Case 3: p = $s+\delta-1$

Since $| r_{j,p-\delta}| \le \left\lfloor \frac{\beta^{\delta-1}}{2} \right\rfloor \times \beta + b - 1.$

Case 3.1: $\delta = 1$

It is obtained that $| r_{j,p-\delta}| \le b - 1$, then the Rewrite statement is valid.

Case 3.2: $\delta = 2$

It is obtained that | $r_{j,p-\delta}$| ≤ $b \times \beta + b - 1$, then the Rewrite statement is also valid.

Correctness:

It is to show that the algorithm gives the correct answer. The expected result should be equal to $\sum_{k=t}^{s}(v_{j,k} + w_{j,k})\beta^k$, for all j = 0, 1, and 2. Consider

$$\sum_{k=t}^{s}(v_{j,k} + w_{j,k})\beta^k = \sum_{k=t}^{t-\delta+1}(v_{j,k} + w_{j,k})\beta^k + \sum_{k=t-\delta}^{s}(v_{j,k} + w_{j,k})\beta^k$$

$$= \left(\sum_{k=t}^{t-\delta+1}(v_{j,k} + w_{j,k})\beta^{k-t+\delta-1}\right) \times \beta^{t-\delta+1} + \sum_{k=t-\delta}^{s}(v_{j,k} + w_{j,k})\beta^k$$

$$= r_{j,t-\delta} \times \beta^{t-\delta+1} + \sum_{k=t-\delta}^{s}(v_{j,k} + w_{j,k})\beta^k$$

$$= r_{j,t-\delta} \times \beta^{t-\delta+1} + \sum_{k=t-\delta}^{s}(s_{j,k})\beta^k$$

$$= r_{j,t-\delta} \times \beta^{t-\delta+1} + \sum_{k=t-\delta}^{s}(c_{j,k}\beta + s_{j,k})\beta^k$$

$$= r_{j,t-\delta} \times \beta^{t-\delta+1} + (c_{j,t-\delta}\beta + s_{j,t-\delta})\beta^{t-\delta} + \sum_{k=t-\delta-1}^{s}(c_{j,k}\beta + s_{j,k})\beta^k$$

$$= (r_{j,t-\delta} + c_{j,t\delta})\beta^{t-\delta+1} + s_{j,t-\delta}\beta^{t-\delta} + \sum_{k=t-\delta-1}^{s}(c_{j,k}\beta + s_{j,k})\beta^k$$

Since $r_{j,t-\delta} + c_{j,t\delta} = z_{j,t}\beta^{\delta-1} + r_{j,t-\delta-1} - \frac{s_{j,t-\delta}}{\beta}$, then

$$\sum_{k=t}^{s}(v_{j,k} + w_{j,k})\beta^k = (z_{j,t}\beta^{\delta-1} + r_{j,t-\delta-1} - \frac{s_{j,t-\delta}}{\beta})\beta^{t-\delta+1} + s_{j,t-\delta}\beta^{t-\delta}$$

$$+ \sum_{k=t-\delta-1}^{s}(c_{j,k}\beta + s_{j,k})\beta^k$$

$$= z_{j,t}\beta^t + r_{j,t-\delta-1}\beta^{t-\delta+1} + \sum_{k=t-\delta-1}^{s}(c_{j,k}\beta + s_{j,k})\beta^k \ .$$

Then we will obtain that

$$\sum_{k=t}^{s}(v_{j,k}+w_{j,k})\beta^{k} \quad =\sum_{k=t}^{s+\delta} z_{j,k}\beta^{k} + r_{j,s+\delta-1}.$$

In the case that $\delta = 1$, $r_{j,s+\delta-1}$ is rewritten into $z_{s+\delta-1}\beta^{\delta-1}$ and in the case where $\delta = 2$, $r_{j,s+\delta-1}$ is rewritten into $z_{s+\delta-1}\beta^{\delta-1} + z_{s+\delta-2}\beta^{\delta-2}$. In the both cases, it is obtained that the result is $\sum_{k=t}^{s} z_{j,k}\beta^{k}$. The proof is completed.

Complexity:

For the addition operation algorithm, the complexity on which we're interested in such as time and space can be described as follows:

1. Time
   - It takes $\Theta(n)$ to do the addition while n is the number of digits.
2. Space
   - The space required for addition operation is $\Theta(\beta)$ while β is a base number.

Example 4.1

For base 5, the digit set is {-3, -2, -1, 0, 1, 2, 3} to do addition calculation between 60i + 23j + 32k and 48i + 36j + 19k.

Solution

Table 4.1 shows an on-line addition using the algorithm above.

Table 4.1 On-line addition of 60i + 23j + 32k and 48i + 36j + 19k

	i	*j*	*k*	*i*	*j*	*k*	*i*	*j*	*k*	*i*	*j*	*k*	
61*i* + 23*j* + 32*k*	0	0	0	2	1	1	2	0	1	1	-2	2	
48*i* + 36*j* + 19*k*	0	0	0	2	1	1	-1	2	-1	3	1	-1	+
	0	0	0	4	2	2	1	2	0	4	-1	1	
Remainder				-1	2	2	1	2	0	-1	-1	1	
Carry digit	1	0	0	0	0	0	1	0	0				
	1	**0**	**0**	**-1**	**2**	**2**	**2**	**2**	**0**	**-1**	**-1**	**1**	

The result is Z =((1, 0, 0)(-1, 2, 2)(2, 2, 0)(-1, -1, 1))$_5$

i-column = (1, -1, 2, -1) = 109

j-column = (0, 2, 2, -1) = 59

k-column = (0, 2, 0, 1) = 51

The result is then 109i + 59j + 51k.

4.2 On-line subtraction

For on-line subtraction algorithm of three-dimensional vector system, the on-line addition algorithm is used to calculate by attaching the negative sign to the second operand.

Theorem 4.2

Given two three-dimensional vectors $V = (v_0, v_1, v_2)$ *and* $W = (w_0, w_1, w_2)$ *where* v_0, v_1, v_2, w_0, w_1, *and* w_2 *are real numbers. Let the signed-vector-digit representation in base* β *with digits in D of V and W be*

$$V = ((v_{0,t}, v_{1,t}, v_{2,t})(v_{0,t-1}, v_{1,t-1}, v_{2,t-1})(v_{0,t-2}, v_{1,t-2}, v_{2,t-2})\ldots(v_{0,s}, v_{1,s}, v_{2,s}))_\beta$$

and

$$W = ((w_{0,t}, w_{1,t}, w_{2,t})(w_{0,t-1}, w_{1,t-1}, w_{2,t-1})(w_{0,t-2}, w_{1,t-2}, w_{2,t-2})\ldots(w_{0,s}, w_{1,s}, w_{2,s}))_\beta,$$

where $D = \{(x, y, z) \mid -b \le x, y, z \le b\}$ *and b is an integer such that* $\beta/2 \le b \le \beta-1$. *The on-line subtraction of W from V can be performed by an on-line addition of V and -W with an on-line delay* δ,

$$\delta = \begin{cases} 2 & : b = \beta/2 \\ 1 & : otherwise \end{cases}.$$

Proof

The proof is obvious by applying Theorem 4.1.

Example 4.2

For base 5, the digit set is {-3, -2, -1, 0, 1, 2, 3} to do subtraction calculation between 50i + 13j + 22k and 38i + 26j + 9k.

Solution

Table 4.2 shows the computation corresponding to the on-line addition algrithm. The result from the algorithm is Z =((0, 0, 0)(1, -1, 0)(-1, 2, 3)(2, 2, -1))$_5$

i-column = (0, 1, -1, 2) = 22

j-column = (0, -1, 2, 2) = -13

k-column = (0, 0, 3, -1) = 14

The result is 22i -13j + 14k.

Table 4.2 On-line subtraction of *50i + 13j + 22k and 38i + 26j + 9k*

	i	*j*	*K*	*i*	*j*	*k*	*i*	*j*	*k*	*i*	*j*	*k*	
60*i* + 13*j* + 22*k*	0	0	0	2	0	1	2	2	-1	0	3	2	
-(38*i* + 26*j* + 8*k*)	0	0	0	-2	-1	0	2	0	-2	2	-1	2	+
	0	0	0	0	-1	1	4	2	-3	2	2	4	
Remainder				0	-1	1	-1	2	2	2	2	-1	
Carry digit	0	0	0	1	0	-1	0	0	1				
	0	**0**	**0**	**1**	**-1**	**0**	**-1**	**2**	**3**	**2**	**2**	**-1**	

4.3 On-line multiplication

The algorithm below illustrates the on-line multiplication algorithm for three-dimensional vector system.

Theorem 4.3

Given two three-dimensional vectors $V = (v_0, v_1, v_2)$ *and* $W = (w_0, w_1, w_2)$ *where* v_0, v_1, v_2, w_0, w_1, *and* w_2 *are real numbers. Let the signed-vector-digit representation in base* β *with digits in D of V and W be*

$$V = ((v_{0,t}, v_{1,t}, v_{2,t})(v_{0,t-1}, v_{1,t-1}, v_{2,t-1})(v_{0,t-2}, v_{1,t-2}, v_{2,t-2})\ldots(v_{0,s}, v_{1,s}, v_{2,s}))_\beta$$

and

$$W = ((w_{0,t}, w_{1,t}, w_{2,t})(w_{0,t-1}, w_{1,t-1}, w_{2,t-1})(w_{0,t-2}, w_{1,t-2}, w_{2,t-2})\ldots(w_{0,s}, w_{1,s}, w_{2,s}))_\beta,$$

where $D = \{(x, y, z) \mid -b \le x, y, z \le b\}$ *and b is an integer such that* $\beta/2 \le b \le \beta-1$. *The on-line multiplication of V and W, denoted by Z, can be performed with an on-line delay* δ *where*

$$\delta = \begin{cases} 2: \beta \ge 3 \\ 3\ \beta = 2 \end{cases}.$$

Proof

The proof of the theorem is given by introducing an algorithm for on-line multiplication in base β with an on-line delay δ. To simplify the proof, let us assume that t = –1. A on-line delay δ can be replaced by introducing the input operand bound (*i.e.*, each input operand must be less than or equal to $\left|\frac{1}{\beta^{\delta}}\right|$ or the first δ digits must be all zero.)

Algorithm: M_V

Input:

$V=((v_{0,t},v_{1,t},v_{2,t})(v_{0,t-1},v_{1,t-1},v_{2,t-1})(v_{0,t-2},v_{1,t-2},v_{2,t-2})\ldots(v_{0,s},v_{1,s},v_{2,s}))_{\beta}$

$W=((w_{0,t},w_{1,t},w_{2,t})(w_{0,t-1},w_{1,t-1},w_{2,t-1})(w_{0,t-2},w_{1,t-2},w_{2,t-2})\ldots(w_{0,s},w_{1,s},w_{2,s}))_{\beta}$

Output:

$Z=((z_{0,t+\delta-1},z_{1,t+\delta-1},z_{2,t+\delta-1})(z_{0,t+\delta-2},z_{1,t+\delta-2},z_{2,t+\delta-2})\ldots(z_{0,2s},z_{1,2s},z_{2,2s}))_{\beta}$

begin

<u>Initialization process</u>

j := 0;

p := -1;

q_j := 0;

r_j := 0;

U_j := 0;

<u>Iteration process</u>

while $p \geq s$ **do**

j := 0;

while $j \leq 2$ **do**

$q_j := q_j + (v_{j,p} \times \beta^{p})$;

$new\_r_j := r_j + (w_{j,p} \times \beta^{p})$;

$j := j + 1$;

enddo

j := 0;

while $j \leq 2$ **do**

$m := (j + 1) \bmod 3;$

$n := (j + 2) \bmod 3;$

$U_j := U_j + (q_m \times w_{n,p}) - (q_n \times w_{m,p})$
$+ (r_m \times v_{n,p}) - (r_n \times v_{m,p}) ;$

$j := j + 1;$

enddo

$j := 0;$

while $j \leq 2$ **do**

$z_{j,p}$ **:=** Integer(U_j) ;

$U_j := (U_j - (z_{j,p})) \times \beta ;$

$r_j := new\_r_j ;$

$j := j + 1;$

enddo

$p := p - 1;$

enddo

<u>Termination process</u>

$j := 0;$

while $j \leq 2$ **do**

Rewrite $U_j := \sum_{k=s-1}^{2s} z_k \beta^{k-s}$;

$j := j + 1;$

enddo

end

Proof of the algorithm

In order to prove that the above algorithm is correct, we will show that the algorithm is valid and correct. Moreover, the complexity of the algorithm will be shown as well.

Validation

It is to show that the result of the multiplication operation of 2 three-dimensional vectors is always in the digit set D as its operands.

By the same way as the proof of an on-line addition algorithm, we have to prove that for any signed-vector-digit $(z_{1,k}, z_{2,k}, z_{3,k})$, all digits $z_{1,k}, z_{2,k}, z_{3,k}$ are elements in the digit set $E = \{ e \mid -b \le e \le b \}$.

First of all, Algorithm M_V can be separated into three steps as follows:

1. Initialization: compute the first partial product from all $\delta - 1$ first digits of the inputs.
2. Iteration: for each input digit,
   a. compute an additional partial product from each operand,
   b. modify the partial product by two additional partial products,
   c. produce the output digit.
3. Termination: produce the output digits from the rest partial product.

From Algorithm M_V, Fig.4.2 shows the concept of an on-line multiplication algorithm. To simplify the proof, let us assume that $t = -1$.

Initialization process

From the algorithm,

(q_0, q_1, q_2)	denotes an input operand V in process,
(r_0, r_1, r_2)	denotes an input operand W in process,
(U_0, U_1, U_2)	denotes the partial product.

Iteration process

For each input digit, $(v_{0,p}, v_{1,p}, v_{2,p})$ and $(w_{0,p}, w_{1,p}, w_{2,p})$, the additional partial product is composed of two parts, as follows:

$$((q_0, q_1, q_2) + (v_{0,p}, v_{1,p}, v_{2,p}) \times \beta^p\ ) \times (w_{0,p}, w_{1,p}, w_{2,p}) \qquad (4.10)$$

and

$$(r_0, r_1, r_2) \times (v_{0,p}, v_{1,p}, v_{2,p}). \qquad (4.11)$$

Note that it is considered only a current input digit of V but not the one of W.

Using the *right-hand finger concept*, (4.10) can be expressed for each j = 0, 1, and 2, as

$$(q_m \times w_{n,p}) - (q_{n,p} \times w_{m,p}),$$

with $m = (j + 1) \bmod 3$ and $n = (j + 2) \bmod 3$, where

$$q_j = q_j + (v_{j,p} \times \beta^p) \tag{4.12}$$

From the algorithm and (4.12), it is

$$q_j \le \left| \frac{1}{\beta^\delta} \right|. \tag{4.13}$$

With the same reason, (4.11) can be expressed for each j = 0, 1, and 2 by

$$(r_m \times v_{n,p}) - (r_n \times v_{m,p}),$$

with $m = (j + 1) \bmod 3$ and $n = (j + 2) \bmod 3$, where

$$r_j = r_j + (w_{j,p} \times \beta^p). \tag{4.14}$$

From the algorithm and (4.14), it is

$$r_j \le \left| \frac{1}{\beta^\delta} \right|. \tag{4.15}$$

Since $z_{j,p}$ **:=** Integer(U_j), and $z_{j,p}$ must be in E, then

$$\left| U_j \right| \le b + \frac{1}{2}. \tag{4.16}$$

The current partial product is then,

$$U_j = U_j + (q_m \times w_{n,p}) - (q_n \times w_{m,p})$$
$$+ (r_m \times v_{n,p}) - (r_{n,p} \times v_{m,p}). \tag{4.17}$$

From $U_j := (U_j - (z_{j,p})) \times\beta$, (4.9), (4.13), (4.15), (4.16), and (4.17), the following condition should be satisfied,

$$\left(\frac{1}{2} \times \beta \right) + \frac{4b}{\beta^\delta} < b + \frac{1}{2}. \tag{4.18}$$

In the case that $\beta = 2$, that is $b = 1$, then the minimum δ is 3. In the case where $\beta \ge 3$, the minimum δ is 2.

Termination process

Since $\left| U_j \right| \le b + \frac{1}{2}$ and $U_j := (U_j - (z_{j,p})) \times\beta$, this can include that $\left| U_j \right| \le \frac{1}{2}$. It is clear that U_j can be rewritten by $\sum_{k=s-1}^{2s1} z_k \beta^k$ where z_k are digits in the set E.

Correctness

The product of two three-dimensional vectors, *V* and *W*, can be computed as follows;

$$V \times W = \sum_{k=t}^{s} V \times (w_{0,k}, w_{1,k}, w_{2,k})\beta^k$$

$$= \sum_{k=t}^{s}\left(\sum_{d=t}^{s}(v_{0,d}, v_{1,d}, v_{2.d}) \times (w_{0,k}, w_{1,k}, w_{2,k})\beta^d\right)\beta^k$$

$$= (v_{0,-1}, v_{1,-1}, v_{2.-1}) \times (w_{0,-1}, w_{1,-1}, w_{2,-1}) \times \beta^{2t}$$

$$+ \sum_{u=t-1}^{s}\left(\sum_{k=t-\delta}^{u}\left(\sum_{d=t}^{u}(v_{0,d}, v_{1,d}, v_{2.d}) \times (w_{0,k}, w_{1,k}, w_{2,k})\beta^d\right)\beta^k\right)$$

$$+ \sum_{u=t-\delta}^{s}\left(\sum_{d=t-\delta}^{u-1}\left(\sum_{k=t}^{u}(w_{0,k}, w_{1,k}, w_{2,k}) \times (v_{0,d}, v_{1,d}, v_{2.d})\beta^k\right)\beta^d\right)$$

where $U_j = (v_{0,-1}, v_{1,-1}, v_{2.-1}) \times (w_{0,-1}, w_{1,-1}, w_{2,-1}) \times \beta^{2t}$.

From the algorithm, $z_{j,p}$ = Integer(U_j) and U_j = (U_j – ($z_{j,p}$)) $\times\beta$, then

the old $U_j = z_{j,-1} + \dfrac{U_j}{\beta}$,or

$$V \times W = z_{j,-1} + \frac{U_j}{\beta}$$

$$+ \sum_{u=t-1}^{s}\left(\sum_{k=t-\delta}^{u}\left(\sum_{d=t}^{u}(v_{0,d}, v_{1,d}, v_{2.d}) \times (w_{0,k}, w_{1,k}, w_{2,k})\beta^d\right)\beta^k\right)$$

$$+ \sum_{u=t-\delta}^{s}\left(\sum_{d=t-\delta}^{u-1}\left(\sum_{k=t}^{u}(w_{0,k}, w_{1,k}, w_{2,k}) \times (v_{0,d}, v_{1,d}, v_{2.d})\beta^k\right)\beta^d\right).$$

For each iteration, one output is produced. This also shows that the degree of the next output is decreased by one (*i.e.*, $\dfrac{U_j}{\beta}$).

Since the value of U_j is bound, an algorithm runs until the last digit of each input operand is accumurated. The last U_j is rewritten into $\sum_{k=s-1}^{2s} z_k\beta^{k-s}$, then the result of the algorithm is as follows:

$$V \times W = \sum_{k=-1}^{s} z_k\beta^k + \frac{U_j}{\beta^s} = \sum_{k=-1}^{s} z_k\beta^k + \frac{\sum_{k=s-1}^{2s} z_k\beta^{k-s}}{\beta^s} = \sum_{k=-1}^{2s} z_k\beta^k .$$

The proof is then completed.

Complexity:

For the multiplication operation algorithm, the complexity on which we're interested in such as time and space can be described as follows:

1. Time
   - It takes $O(n^2)$ to do the multiplication while n is the number of digits.
2. Space
   - The space required for multiplication operation is $O(n)$ while n is a base number.

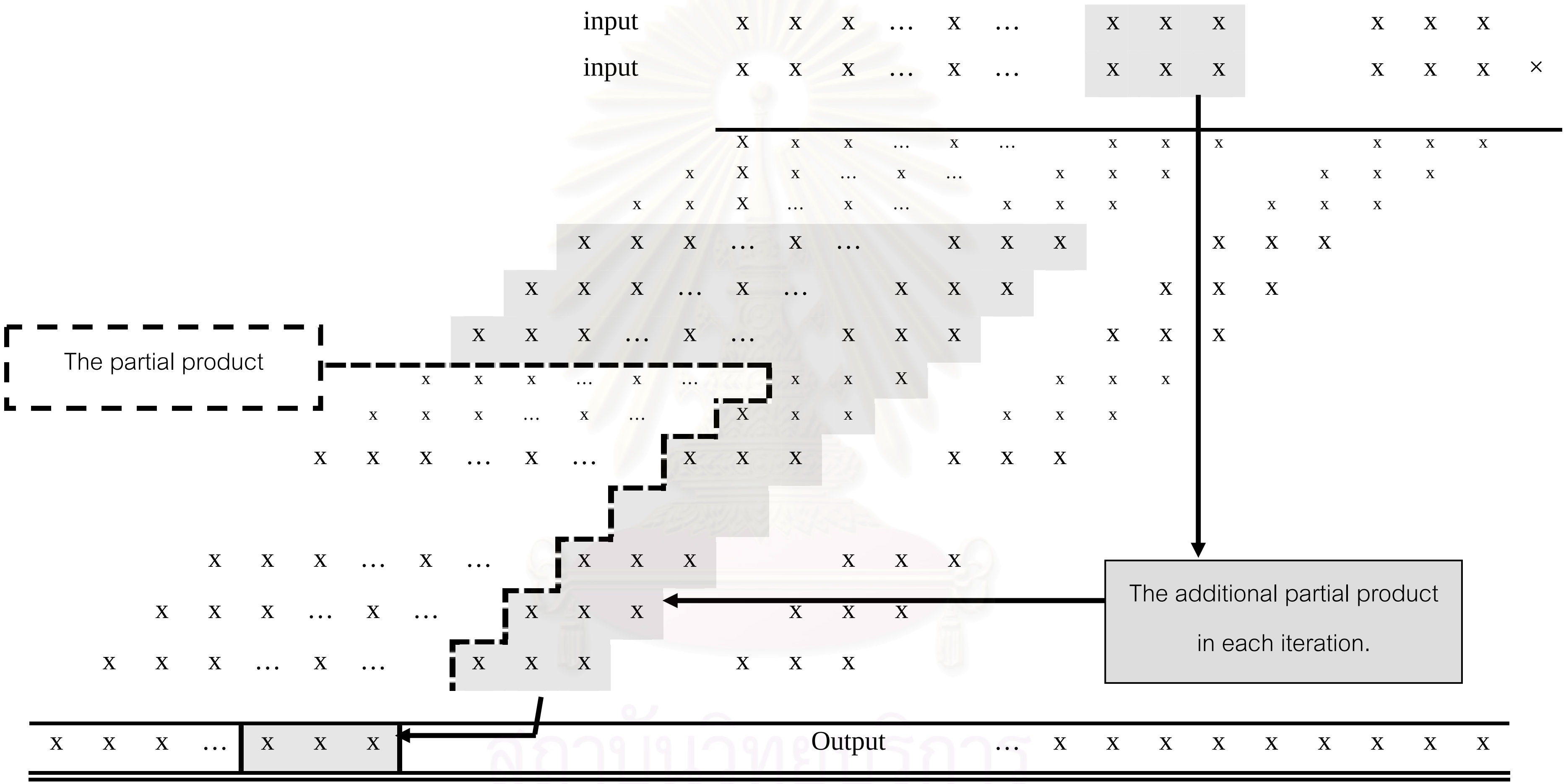

Figure. 4.2 The process of an on-line multiplication algorithm M_V

Example 4.3

For base 3, the digit set is {-2,-1, 0, 2, 1} to do multiplication calculation between V and W while V=84i + 27j +13k and W=37i +93j + 82k.

Solution

The representation of V and W are as follows:

V = ((1, 0, 0) (0, 1, 0) (0, 0, 1) (1, 0, 1) (0, 0, 1))$_3$

W = ((0, 1, 1) (1, 0, 0) (1, 1, 0) (0, 1, 0) (1, 0, 1))$_3$

Since an on-line delay is 2, the operand bound is $\frac{1}{3^2}$

Let $V' = V \times \beta^{-7}$.

The representation of V' = (.(1, 0, 0) (0, 1, 0) (0, 0, 1) (1, 0, 1) (0, 0, 1))$_3$

Let $W' = W \times \beta^{-7}$.

The representation of W' = (.(0, 1, 1) (1, 0, 0) (1, 1, 0) (0, 1, 0) (1, 0, 1))$_3$

Fig 4.3 shows the process of the classical multiplication which gives the result at the end as follows:

i-column = 1-1-1-2-1-2-10 = 1005

j-column = -100012201 = -6407

k-column = 1001001000 = 6813

That gives the result to be 1005i - 6407j + 6813k

Table 4.3 shows the process of the multiplication algorithm (M_v).

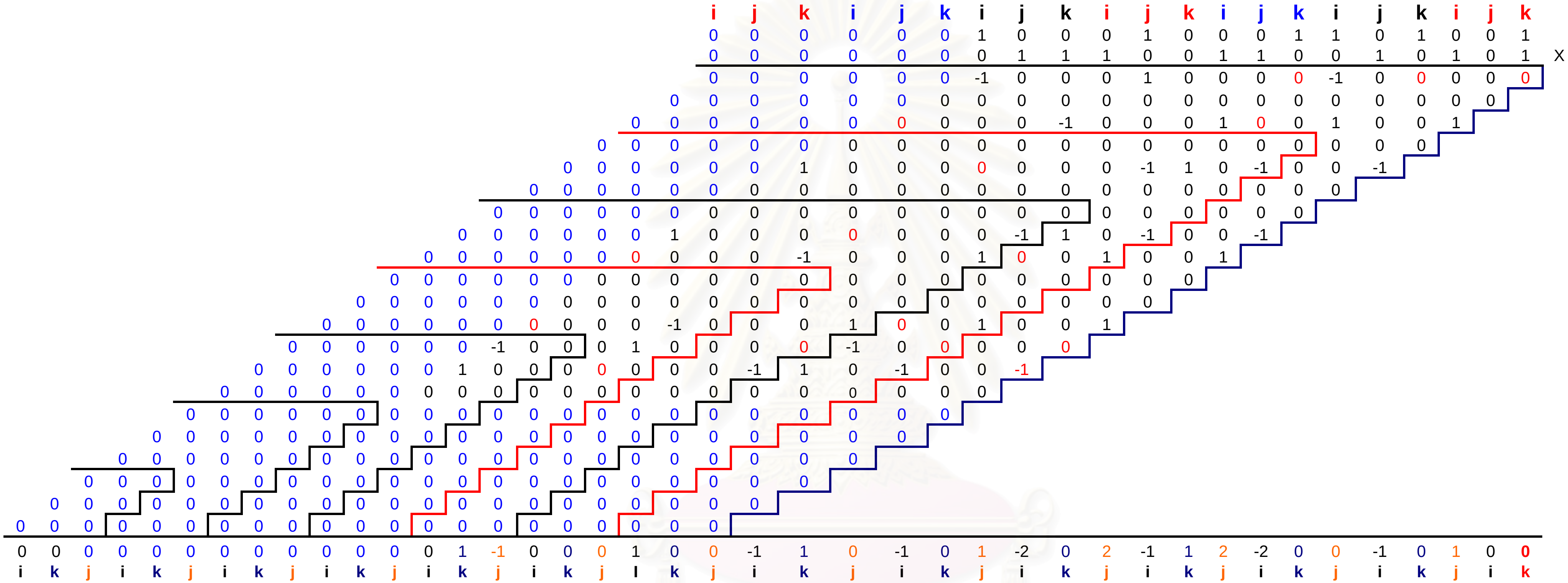

Figure 4.3 On-line multiplication of *84i + 27j + 13k and 37i + 93j + 82k*

Following the algorithm M_v,

Table 4.3 On-line multiplication of *84i + 27j + 13k and 37i + 93j + 82k following the algorithm* M_v

Step	***V***	***W***	***Q***	***r***	***U***	***z***	***New U***
1	(0,0,0)	(0,0,0)	0	0	0	(0,0,0)	0
2	(0,0,0)	(0,0,0)	0	0	0	(0,0,0)	0
3	(1,0,0)	(0,1,1)	$\frac{1}{27}i$	$\frac{1}{27}j+\frac{1}{27}k$	$\frac{-1}{27}j+\frac{1}{27}k$	(0,0,0)	$\frac{-1}{9}j+\frac{1}{9}k$
4	(0,1,0)	(1,0,0)	$\frac{1}{27}i+\frac{1}{81}j$	$\frac{1}{81}i+\frac{1}{27}j+\frac{1}{27}k$	$\frac{1}{27}i+\frac{-1}{9}j+\frac{8}{81}k$	(0,0,0)	$\frac{1}{9}i+\frac{-1}{3}j+\frac{8}{27}k$
5	(0,0,1)	(1,1,0)	$\frac{1}{27}i+\frac{1}{81}j+\frac{1}{243}k$	$\frac{4}{243}i+\frac{10}{243}j+\frac{1}{27}k$	$\frac{17}{243}i+\frac{-77}{243}j+\frac{26}{81}k$	(0,0,0)	$\frac{17}{81}i+\frac{-77}{81}j+\frac{26}{27}k$
6	(1,0,1)	(0,1,0)	$\frac{28}{729}i+\frac{1}{81}j+\frac{4}{729}k$	$\frac{4}{243}i+\frac{31}{729}j+\frac{1}{27}k$	$\frac{119}{729}i+\frac{-236}{243}j+\frac{760}{729}k$	(0,-1,1)	$\frac{119}{243}i+\frac{7}{81}j+\frac{31}{243}k$
7	(0,0,1)	(1,0,1)	$\frac{28}{729}i+\frac{1}{81}j+\frac{13}{2187}k$	$\frac{37}{2187}i+\frac{31}{729}j+\frac{82}{2187}k$	$\frac{335}{729}i+\frac{154}{2187}j+\frac{28}{243}k$	(0,0,0)	$\frac{335}{243}i+\frac{154}{729}j+\frac{28}{81}k$

Step	***V***	***W***	***Q***	***r***	***U***	***Z***	***U***
					$\frac{335}{243}i+\frac{154}{729}j+\frac{28}{81}k$	(1,0,0)	$\frac{92}{81}i+\frac{154}{243}j+\frac{28}{27}k$
9					$\frac{92}{81}i+\frac{154}{243}j+\frac{28}{27}k$	(1,1,1)	$\frac{11}{27}i+\frac{-89}{81}j+\frac{1}{9}k$
10					$\frac{11}{27}i+\frac{-89}{81}j+\frac{1}{9}k$	(0,-1,0)	$\frac{11}{9}i+\frac{-8}{27}j+\frac{1}{3}k$
11					$\frac{11}{9}i+\frac{-8}{27}j+\frac{1}{3}k$	(1,0,0)	$\frac{2}{3}i+\frac{-8}{9}j+1k$
12					$\frac{2}{3}i+\frac{-8}{9}j+1k$	(1,-1,1)	$\bar{1}i+\frac{1}{3}j$
13					$\bar{1}i+\frac{1}{3}j$	(-1,0,0)	j
14					j	(0,1,0)	0

The result from the algorithm is $Z^{'}$ = (.(0,0,0) (0,0,0) (0,0,0) (0,0,0) (0,0,0) (0,-1,1) (0,0,0) (1,0,0) (1,1,1) (0,-1,0) (1,0,0) (1,-1,1) (-1,0,0) (0,1,0))

i-column = .000000011011-10 $\times 3^{14}$ = 11011-10 = 1005

j-column = .00000-1001-10-101 $\times 3^{14}$ = -1001-10-101 = -6407

k-column = .00000100100100 $\times 3^{14}$ = 100100100 = 6813

Therefore, the result is 1005i -6407j + 6813k

CHAPTER V

CONCLUSION

In serial computation, fundamental arithmetic calculations such as addition, subtraction and multiplication are usually processed in LSDF (least significant digit first) mode meaning that the calculation is taken place from the rightmost to the leftmost digit while the division operation is performed in the MSDF (most significant digit first) calculation mode. The pipeline concept has been brought up to apply for the further idea on how the output for each digit can be produced while the calculation of the rest digit can continue. In fact, the output can be produced without waiting the whole calculation to be done.

Three dimensional vector systems have been applied to current technology such as fluid mechanic, mechanical engineering, electrical engineering, etc. Therefore, the time complexity for doing any calculation is very crucial.

In this work, the *signed-vector-digit representation* is proposed in order to represent three-dimensional vectors. Three numbers from all dimensions of three-dimensional vectors are combined. Therefore, they will not be separately maintained anymore. With the normal means of cross product calculation, that would take nine times of multiplication and another eight times of addition in order to achieve one cross product of two three-dimensional vectors. With the signed-vector-digit representation, for doing the cross product of any two vectors, it takes only one real number slightly modified classical multiplication as illustrated by Figure 4.2.

The advantage of using the on-line arithmetic concept is to get the output for each digit produced without waiting for the whole calculation to be completed. With the newly introduced signed-vector-digit representation and on-line fundamental arithmetic algorithms for three-dimensional vector system which are addition, subtraction, and multiplication (cross product), the classical computation can be applied as a result of that, the complexity of the computation for signed-vector-digit would be less compared to the three-dimensional vector format.

REFERENCES

[1] W. Penney. A binary system for complex numbers. *JACM* 12 (1965): 247-248.

[2] T. Safer. Polygonal radix representations of complex numbers. *Theoretical Computer Science* 210 (1999): 159-171.

[3] J. Stewart. *Calculus (Early Trancendentals)*. 3rd Ed. CA: Brooks/Cole, Pacific Grove, 1995.

[4] K.S. Trivedi and M.D. Ercegovac. On-line algorithms for division and multiplication. *IEEE Transactions on Computers* 26 (1977): 681-687.

[5] A. Avizienis. Signed-digit number representations for fast parallel arithmetic. *IRE Transactions on Electronic Computers* 10 (1961): 389-400.

[6] B. Parhami. Generalized Signed-Digit Number Systems: A Unifying Framework for Redundant Number Representations. *IEEE Transactions on Computers* 39 (1990): 89-98.

[7] B. Parhami. *Computer Arithmetic: Algorithms and Hardware Designs*. London: Oxford University Press, 2000.

[8] J. W. Joiner. *A History of Vector Analysis* (doctoral dissertation at George Peabody College for Teachers), 1971.

[9] A. Surarerks, Digit Set Conversion by On-Line Finite Automata. *Bullentin of the Belgian Mathematical Society* 8 (2001): 337-258.

[10] Ch. Frougny and A. Surarerks. On-line multiplication in real and Complex base. *Proceedings of the 16th IEEE Symposium on Computer Arithmetic* (2003): 212-219.

[11] D. E. Knuth. An Imaginary Number System. *Communications of the ACM* (1960): 245-247.

BIOGRAPHY

Name	Saravut Rangsunvigit
Gender	Male
Date of Birth	December 9,1975
Marital Status	Single

Education History

1999-2002	B.Sc in Computer Science, University of Manitoba (Manitoba, Canada)
1993-1997	B.Eng in Civil Engineering, Kasetsart University

Employment History

October 2002-Present	Reuters Software(THAILAND) Ltd.
April 2002-September 2002	Canada Safeway Co.,Ltd.
September 2001-January 2002	Qunara Co.,Ltd (A Division of Mantioba Telecom Services)
January 2001-May 2001	Manitoba Education and Training